Terms used ipc virtualization

Found 1,622 of 198,617

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown
Relevance scale □ ▂ ▄ ▆ ■

1 Migration: The design and implementation of Zap: a system for migrating computing environments ■
Steven Osman, Dinesh Subhraveti, Gong Su, Jason Nieh
December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI
Publisher: ACM Press
Full text available: pdf(2.06 MB) Additional Information: full citation, abstract, references, citings

We have created Zap, a novel system for transparent migration of legacy and networked applications. Zap provides a thin virtualization layer on top of the operating system that introduces pods, which are groups of processes that are provided a consistent, virtualized view of the system. This decouples processes in pods from dependencies to the host operating system and other processes on the system. By integrating Zap virtualization with a checkpoint-restart mechanism, Zap can migrate a pod of p ...

2 Security and eliability: A feather-weight virtual machine for windows applications ■
Yang Yu, Fanglu Guo, Susanta Nanda, Lap-chung Lam, Tzi-cker Chiueh
June 2006 **Proceedings of the second international conference on Virtual execution environments VEE '06**
Publisher: ACM Press
Full text available: pdf(192.18 KB) Additional Information: full citation, abstract, references, index terms

Many fault-tolerant and intrusion-tolerant systems require the ability to execute unsafe programs in a realistic environment without leaving permanent damages. Virtual machine technology meets this requirement perfectly because it provides an execution environment that is both realistic and isolated. In this paper, we introduce an OS level virtual machine architecture for Windows applications called *Feather-weight Virtual Machine* (FVM), under which virtual machines share as many resources ...

Keywords: copy on write, mobile code security, namespace virtualization, system call interception, virtual machine

3 Service infastructure and network management: MobiDesk: mobile virtual desktop computing ■
Ricardo A. Baratto, Shaya Potter, Gong Su, Jason Nieh
September 2004 **Proceedings of the 10th annual international conference on Mobile computing and networking MobiCom '04**
Publisher: ACM Press
Full text available: pdf(580.39 KB) Additional Information: full citation, abstract, references, citings, index terms

We present MobiDesk, a mobile virtual desktop computing hosting infrastructure that leverages continued improvements in network speed, cost, and ubiquity to address the complexity, cost, and mobility limitations of today's personal computing infrastructure. MobiDesk transparently virtualizes a user's computing session by abstracting underlying system resources in three key areas: display, operating system, and network. It provides a thin virtualization layer that decouples a user's computing ses ...

Keywords: computer utility, network mobility, on-demand computing, process migration, thin-client computing, virtualization

4 Improving the browsing experience: WebPod: persistent Web browsing sessions with pocketable storage devices
Shaya Potter, Jason Nieh
May 2005 **Proceedings of the 14th international conference on World Wide Web WWW '05**
Publisher: ACM Press
Full text available: pdf(166.59 KB) Additional Information: full citation, abstract, references, index terms

We present WebPod, a portable system that enables mobile users to use the same persistent, personalized web browsing session on any Internet-enabled device. No matter what computer is being used, WebPod provides a consistent browsing session, maintaining all of a user's plugins, bookmarks, browser web content, open browser windows, and browser configuration options and preferences. This is achieved by leveraging rapid improvements in capacity, cost, and size of portable storage devices. WebPod p ...

Keywords: checkpoint/restart, portable storage, process migration, virtualization, web browsing

5 Queue Focus: Building Systems to Be Shared, Securely
Poul-Henning Kamp, Robert Watson
July 2004 **Queue**, Volume 2 Issue 5
Publisher: ACM Press
Full text available: pdf(575.43 KB) html(41.37 KB) Additional Information: full citation, citings, index terms

6 An operating system approach to securing e-services
Chris Dalton, Tse Huong Choo
February 2001 **Communications of the ACM**, Volume 44 Issue 2
Publisher: ACM Press
Full text available: pdf(132.65 KB) html(34.72 KB) Additional Information: full citation, references, citings, index terms

7 The performance of μ-kernel-based systems
Hermann Härtig, Michael Hohmuth, Jochen Liedtke, Sebastian Schönberg, Jean Wolter
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5
Publisher: ACM Press
Full text available: pdf(2.02 MB) Additional Information: full citation, references, citings, index terms

8 Improving the reliability of commodity operating systems
Michael M. Swift, Brian N. Bershad, Henry M. Levy
February 2005 **ACM Transactions on Computer Systems (TOCS)**, Volume 23 Issue 1
Publisher: ACM Press
Full text available: pdf(459.98 KB) Additional Information: full citation, abstract, references, citings, index terms

Despite decades of research in extensible operating system technology, extensions such as device drivers remain a significant cause of system failures. In Windows XP, for example, drivers account for 85&percent; of recently reported failures.This article describes Nooks, a *reliability subsystem* that seeks to greatly enhance operating system (OS) reliability by isolating the OS from driver failures. The Nooks approach is practical: rather than guaranteeing complete fault tolerance through ...

Keywords: I/O, Recovery, device drivers, protection, virtual memory

9 Improving the reliability of commodity operating systems
Michael M. Swift, Brian N. Bershad, Henry M. Levy
October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5
Publisher: ACM Press
Full text available: pdf(262.78 KB) Additional Information: full citation, abstract, references, citings, index terms

Despite decades of research in extensible operating system technology, extensions such as device drivers remain a significant cause of system failures. In Windows XP, for example, drivers account for 85% of recently reported failures. This paper describes Nooks, a *reliability subsystem* that seeks to greatly enhance OS reliability by isolating the OS from driver failures. The Nooks approach is practical: rather than guaranteeing complete fault tolerance through a new (and incompatible) OS ...

Keywords: I/O, device drivers, protection, recovery, virtual memory

10 Testbed directions and experience: PlanetLab: an overlay testbed for broad-coverage services
Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, Mic Bowman
July 2003 **ACM SIGCOMM Computer Communication Review**, Volume 33 Issue 3
Publisher: ACM Press
Full text available: pdf(158.92 KB) Additional Information: full citation, abstract, references, citings

PlanetLab is a global overlay network for developing and accessing broad-coverage network services. Our goal is to grow to 1000 geographically distributed nodes, connected by a disverse collection of links. PlanetLab allows multiple service to run concurrently and continuously, each in its own slice of PlanetLab. This paper discribes our initial implementation of PlanetLab, including the mechanisms used to impelment virtualization, and the collection of core services used to manage PlanetLab.

11 Microkernels and virtual machines: Reducing TCB size by using untrusted components: small kernels versus virtual-machine monitors
Michael Hohmuth, Michael Peter, Hermann Härtig, Jonathan S. Shapiro
September 2004 **Proceedings of the 11th workshop on ACM SIGOPS European workshop: beyond the PC EW11**
Publisher: ACM Press
Full text available: pdf(82.24 KB) Additional Information: full citation, abstract, references

Secure systems are best built on top of a small trusted operating system: The smaller the operating system, the easier it can be assured or verified for correctness.In this paper, we oppose the view that virtual-machine monitors (VMMs) are the smallest systems that provide secure isolation because they have been specifically designed to provide little more than this property. The problem with this assertion is that VMMs typically do not support interprocess communication, complicating the use of ...

12 A comparison of software and hardware techniques for x86 virtualization
Keith Adams, Ole Agesen
October 2006 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , Proceedings of the 12th international conference on Architectural support for programming languages and operating systems ASPLOS-XII**, Volume 40 , 34 , 41 Issue 5 , 5 , 11
Publisher: ACM Press

Full text available: pdf(236.66 KB) Additional Information: full citation, abstract, references, index terms

Until recently, the x86 architecture has not permitted classical trap-and-emulate virtualization. Virtual Machine Monitors for x86, such as VMware ® Workstation and Virtual PC, have instead used binary translation of the guest kernel code. However, both Intel and AMD have now introduced architectural extensions to support classical virtualization.We compare an existing software VMM with a new VMM designed for the emerging hardware support. Surprisingly, the hardware VMM often suffers lower ...

Keywords: MMU, SVM, TLB, VT, dynamic binary translation, nested paging, virtual machine monitor, virtualization, x86

13 The operating system kernel as a secure programmable machine
Dawson R. Engler, M. Frans Kaashoek, James W. O'Toole
January 1995 **ACM SIGOPS Operating Systems Review**, Volume 29 Issue 1
Publisher: ACM Press
Full text available: pdf(508.88 KB) Additional Information: full citation, abstract, index terms

To provide modularity and performance, operating system kernels should have only minimal embedded functionality. Today's operating systems are large, inefficient and, most importantly, inflexible. In our view, most operating system performance and flexibility problems can be eliminated simply by pushing the operating system interface lower. Our goal is to put abstractions traditionally implemented by the kernel out into user-space, where user-level libraries and servers abstract the exposed hard ...

14 Operating system: The persistent relevance of the local operating system to global applications
Jay Lepreau, Bryan Ford, Mike Hibler
September 1996 **Proceedings of the 7th workshop on ACM SIGOPS European workshop: Systems support for worldwide applications EW 7**
Publisher: ACM Press
Full text available: pdf(828.93 KB) Additional Information: full citation, abstract, references, citings

The growth and popularity of loosely-coupled distributed systems such as the World Wide Web and the touting of Java-based systems as the solution to the issues of software maintenance, flexibility, and security are changing the research emphasis away from traditional single node operating system issues. Apparently, the view is that traditional OS issues are either solved problems or minor problems. By contrast, we believe that building such vast distributed systems upon the fragile infrastructur ...

15 Operating and runtime systems for high-end computing systems: Performance evaluation of automatic checkpoint-based fault tolerance for AMPI and Charm++
Gengbin Zheng, Chao Huang, Laxmikant V. Kalé
April 2006 **ACM SIGOPS Operating Systems Review**, Volume 40 Issue 2
Publisher: ACM Press
Full text available: pdf(696.92 KB) Additional Information: full citation, abstract, references, index terms

As the size of high performance clusters multiplies, the probability of system failure grows substantially, posing an increasingly significant challenge for scalability. Checkpoint-based fault tolerance methods are effective approaches at dealing with faults. With these methods, the state of the entire parallel application is checkpointed to reliable storage. When a fault occurs, the application is restarted from a recent checkpoint. However, the application developer is required to write signif ...

16 Are virtual-machine monitors microkernels done right?
Gernot Heiser, Volkmar Uhlig, Joshua LeVasseur
January 2006 **ACM SIGOPS Operating Systems Review**, Volume 40 Issue 1
Publisher: ACM Press
Full text available: pdf(124.78 KB) Additional Information: full citation, abstract, references, index terms

A paper by Hand *et al.* at the recent HotOS workshop re-examined microkernels and contrasted them to virtual-machine monitors (VMMs). It found that the two kinds of systems share architectural commonalities but also have a number of technical differences which the paper examined. It concluded that VMMs are a special case of microkernels, "microkernels done right".A closer examination of that paper shows that it contains a number of statements which are poorly justified or even refuted by t ...

17 Xen and the art of virtualization
Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfield
October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5
Publisher: ACM Press
Full text available: pdf(168.76 KB) Additional Information: full citation, abstract, references, citings, index terms

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.This paper presents Xen, an x86 virtual machine monit ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

18 Vertical profiling: understanding the behavior of object-priented applications
Matthias Hauswirth, Peter F. Sweeney, Amer Diwan, Michael Hind
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10
Publisher: ACM Press
Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Object-oriented programming languages provide a rich set of features that provide significant software engineering benefits. The increased productivity provided by these features comes at a justifiable cost in a more sophisticated runtime system whose responsibility is to implement these features efficiently. However, the virtualization introduced by this sophistication provides a significant challenge to understanding complete system performance, not found in traditionally compiled languages ...

Keywords: hardware performance monitors, perturbation, software performance monitors, vertical profiling, whole-system analysis

19 Exokernel: an operating system architecture for application-level resource management
D. R. Engler, M. F. Kaashoek, J. O'Toole
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5
Publisher: ACM Press
Full text available: pdf(2.16 MB) Additional Information: full citation, references, citings, index terms

20 The implementation and evaluation of dynamic code decompression using DISE
Marc L. Corliss, E. Christopher Lewis, Amir Roth
February 2005 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 4 Issue 1
Publisher: ACM Press
Full text available: pdf(1.23 MB) Additional Information: full citation, abstract, references, index terms

Code compression coupled with dynamic decompression is an important technique for both embedded and general-purpose microprocessors. *Postfetch decompression*, in which decompression is performed after the compressed instructions have been fetched, allows the instruction cache to store compressed code but requires a highly efficient decompression implementation. We propose implementing postfetch decompression using a new hardware facility called *dynamic instruction stream editing* (DI ...

Keywords: Code compression, DISE, code decompression, dynamic instruction stream editing, dynamic instrumentation

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next